James Tucker

Invercargill, New Zealand • james@edgedepth.com
github.com/edgedepthhq • linkedin.com/in/james-tucker-667b50a6

New Zealand citizen. Full Australian working rights (SCV 444), no sponsorship required.

Systems engineer. Built and operate a real-time market data and replay platform for crypto perpetual futures: a C++20 / WebAssembly client, a Go ingestion backend, and deterministic tick-level replay over 400+ days of recorded order books. The client is open source. Previously three and a half years as a full-stack engineer in Austria. Computer Science and Statistics, University of Otago.

Experience

Founder and Software Engineer, EdgeDepth — Invercargill, New Zealand — 2025 to present

- Built the browser client in C++20, compiled to WebAssembly with Emscripten and rendered through Dear ImGui, ImPlot and WebGL2, drawing dense real-time market visualisations at up to 180 FPS.
- Split ingest across two synchronisation paths chosen by data semantics: order-book updates land in a mutex-protected write model that the render thread publishes once per frame, because only the latest book is ever drawn, while discrete events drain in order through a dispatch queue under a 3 ms per-frame budget so an ingest burst can never consume a whole frame.
- Implemented deterministic tick-by-tick replay with timeline scrubbing and variable speed, rebuilding the order book, trade tape, liquidations and derived state in the exact configuration they held rather than replaying candles.
- Ran a historical depth backfill of 38.7 billion rows across 171,146 files (337 GiB) in 71 hours with zero errors.
- Built a deterministic research engine over the recording: a closed 39-feature grammar across 844 instruments returning occurrence counts with denominators and forward outcomes. Queries are content-addressed and byte-reproducible, and outcome fields are structurally excluded from predicates, so selecting on the result is impossible by construction rather than by discipline.
- Designed the Go backend and run it as distributed services across bare-metal Linux hosts in three locations joined by WireGuard: a thin producer holding the exchange connections for 558 instruments publishes to NATS JetStream, and analytics consumers reconstruct order books, process concurrent streams, detect events and deliver Protobuf over WebSocket, either side redeployable without data loss.
- Operate all of it: PostgreSQL/TimescaleDB with streaming replication to a standby, tiered storage from a NATS tail through TimescaleDB to Parquet and object storage, backups with a daily automated integrity gate, Prometheus and Grafana, nginx, firewalling, and my own on-call.

Full Stack Software Engineer, WEBPUNKS — Klagenfurt, Austria — Jul 2017 to Dec 2020

- Full-stack engineer in a team of roughly seven, split across backend and front end, building trail-angels.com and its Bookyourtrail booking platform for guided hiking-trail operators across the Alpine-Adriatic region: AngularJS front end, TypeScript and Node.js services, MongoDB.
- Worked to production discipline across three and a half years: git workflow, unit testing, code review, and maintaining a codebase other engineers had to read and extend.

Open Source

- **edgedepth-terminal**: the C++20 / WebAssembly trading terminal above, self-hostable with Docker, with the public protobuf contract, replay-pack reader, and an **architecture document** covering threading, ownership, the wire contract and the invariants contributors must not break.
- **edgedepth-research-mcp**: Model Context Protocol server exposing the deterministic research engine to AI agents.

Technical Skills

- **C++20**: WebAssembly and Emscripten, Dear ImGui / ImPlot, WebGL2, SDL3, real-time render loops
- **Go**: concurrency, actor systems, high-throughput ingestion pipelines
- **Systems and operations**: bare-metal Linux, systemd, Docker, private networking and WireGuard, nginx, TLS, Prometheus and Grafana, CI/CD, incident response, production on-call
- **Distributed data**: PostgreSQL, TimescaleDB, streaming replication, NATS JetStream, Redis, Parquet, S3-compatible object storage and CDN, Protobuf
- **Cloud**: AWS, Azure and Google Cloud; Cloudflare R2 and CDN in production
- **Also**: TypeScript, Node.js, MongoDB, Python; applied statistics and production ML pipelines

Education

Bachelor of Science in Computer Science and Statistics (double major) — University of Otago — 2021 to 2024

- Applied data science, machine learning and mathematics. Coursework included stochastic modelling: Poisson and point processes, Markov chains, hidden Markov models, geostatistics, forecasting and simulation.